



Kandidatarbete i Medieteknik, 30 hp
Vårtermin 2013

Spelutveckling, problem och utmaningar

Problem och utmaningar vid utveckling av tävlingsinriktat gratisspel

John Fristedt

Handledare: Peter Giger & Jonas Svegland
Examinator: Lena Trojer
Blekinge Tekniska Högskola, Sektionen för planering och mediedesign

1. Abstrakt

1.1. Abstrakt sammanfattning av innehåll

Detta arbete ämnar undersöka problem och utmaningar relaterade till design- och produktion av ett tävlingsinriktat spel för gratis utgivning samt utforska potentiella lösningar och designfilosofier genom att påbörja utveckling av ett spel. Problem som uppstår, lösningar som tillämpas och val som görs dokumenteras varpå resultat analyseras.

Undersökningen tar upp design av både spelmekaniska och grafiska element samt produktionen av spelet och dess resurser, saker som är viktiga att tänka på och några saker som kan gå fel.

Nyckelord: Design, produktion, problem, analys, självstudier.

1.2. Abstract summary of content

This production means to examine the problems and challenges related to the design and production of a competitive FRS for free distribution and play, and to explore potential solutions and design philosophies by beginning production on a game. Problems that arise, solutions that are applied and choices that are made will be documented whereupon the results are analyzed.

This examination touches upon the design of both gameplay and graphics within games along with the production of it and related resources, things that are important to keep in mind, and a few things that may not go according to plan.

Keywords: Design, production, problems, analysis, self-education.

Innehållsförteckning

1. Abstrakt	2
1.1. Abstrakt sammanfattning av innehåll	2
1.2. Abstract summary of content.....	2
2. Inledning.....	4
3. Problemområde	5
3.1. Bakgrund.....	5
3.1.1. Spelet och designens natur.....	5
3.1.2. Produktionens fokus.....	6
3.1.3. Distribution och monetarisering.....	6
3.1.4. FPS-spel som tävling.....	7
3.2. Frågeställning	9
3.3. Syfte	10
3.4. Tidigare forskning.....	11
4. Tillvägagångssätt.....	14
4.1. Arbetsmodell	14
4.1.1. Produktionsmål och produktbeskrivning.....	14
4.2. Produktion	16
4.2.1. Produktionens mål.....	16
4.2.2. De stora problemen	16
4.2.3. Första problemet, paketstruktur	17
4.2.4. Andra problemet, analoga kontroller.....	17
4.2.5. Tredje problemet, serverstruktur	17
4.2.6. Fjärde problemet, spelmotorn Sunburn 2.0	18
4.2.7. Alternativet, spelmotorn UDK (Unreal Development Kit)	20
4.3. Design	21
4.3.1. Gameplaytema	21
4.3.2. Grafiskt tema.....	21
4.3.3. Karaktärer	22
4.3.4. Former	22
4.3.5. Färger.....	22
5. Resultat och diskussion	23
5.1. Resultat	23
5.1.1. Produktion.....	23
5.1.2. Design.....	23
5.2. Diskussion	24
5.2.1. Produktion.....	24
5.2.2. Design.....	25
5.3. Slutdiskussion och konklusion	26
6. Ordlista	27
7. Källförteckning	28

2. Inledning

Det finns många saker att tänka på för en nybliven spelutvecklare, många problem, utmaningar, regler, filosofier etc. som man måste ta hänsyn till för att ens spel ska synas och fungera som lämpliga alternativ till andra spel på marknaden. I den här undersökningen tittar jag närmare på problem och utmaningar man ställs inför som en ung spelutvecklare, varpå jag försöker lösa dem eller hitta en teoretisk lösning. Undersökningen relaterar både till design och produktion och problemen jag ställs inför är av både personlig och professionell natur då de utgår ifrån dels beprövade metoder och erfarenhet och dels min egna brist på erfarenhet och de insikter jag når genom att prova mig fram.

3. Problemområde

3.1. Bakgrund

3.1.1. Spelet och designens natur

Speldesign är processen som skapar ett spels innehåll och regler. *God* speldesign är processen som skapar motiverande mål och regler som tvingar spelaren att fatta meningsfulla beslut för att uppnå dessa mål. God speldesign utgår från spelaren. Detta innebär framförallt att ta hänsyn till spelarens och hennes önskemål. Snarare än att introducera regler som tvingar fram ett visst beteende bör man konstruera ett gameplay som i sig motiverar spelaren att gå i den riktning designern önskar. Speldesignens essens är att skapa möjligheter för spelaren att fatta meningsfulla beslut som påverkar spelets utfall. (Brenda Brathwaite och Ian Schreiber 2008, 2)

Vi ser alltså att spelets värld är en av illusioner och regler. Brister i designen eller implementationen kan väldigt lätt bryta inlevelsen och förstöra upplevelsen. Röken och speglarna måste distrahera oss från vad som sker bakom kulisserna. Det är därför väldigt viktigt att de illusioner, som byggs upp i ett spel samspelar på ett sätt, som distraherar spelaren från enskilda element och istället förmedlar en övertygande och trovärdig helhet.

Att konstruera en sådan miljö är en utmaning som bara växer i takt med upplevelsen. Därför finns det mycket att vinna i att identifiera så många utmaningar och problem som möjligt innan produktionen påbörjas, för att i förväg kunna utforma lösningar och strategier.

3.1.2. Produktionens fokus

Produktionsmetodik är av stor vikt för såväl enmansprojekt som större produktioner. När produkten finns beskriven sätts den i produktion, vilket ofta är en tidskrävande process. För att minimera kostnaderna och maximera resultatet måste produktionen ske på ett effektivt sätt. Spel som brukar en bristfällig (eller helt saknar) utvecklingsmetodik kräver ofta mer tid och pengar att utveckla och släpps ofta med onödiga buggar och generellt lägre kvalitet. (Bethke 2003, 3)

Denna undersökning kommer att försöka behandla design- och produktionsproblem som uppstår vid utveckling av ett tävlingsinriktat onlinespel för gratis utgivning. Designområdet är uppdelat i två delar; feedback (grafik och ljud) och spelmekanisk design.

Det mesta av den feedback spelaren får kommer genom de visuella och auditiva kanalerna, det är därför viktigt att allt som spelaren ser och hör är noggrant genomtänkt och väl implementerat. Den spelmekaniska delen skulle kunna beskrivas som det gränssnitt genom vilket spelaren interagerar med spelet och är avgörande för vilken sinnesstämning spelaren ska sättas i. Om spelet ska ha en snabb, smidig och rättvis tävlingsmiljö är det inte optimalt att sträva efter perfekt realism eller otroligt detaljerade karaktärsmodeller och omgivningar. Man kanske till och med sänker detaljnivån i sitt spel för att frambringa specifika element.

3.1.3. Distribution och monetarisering

F2P är en modell som fler och fler spelutvecklare börjat använda och som ser ut att vara ett starkt alternativ till den mer traditionella försäljningen av fysiska kopior eller prenumeration.

Konsumenter attraheras av spel som följer F2P modellen eftersom dessa spel ses som en mindre finansiell risk. Det känns säkrare för användarna och de blir mer villiga att betala små summor för digitala föremål som förbättrar deras spelupplevelse när de väl har bekantat sig med spelet.

(Giuditta De Prato and Jean Paul Simon, 2011)

Vi går dessutom mer och mer in i en digital tidsålder, där konsumenten väldigt ofta föredrar att införskaffa sina spel genom digitala källor. Piratkopiering ses av många som ett stort problem och många större företag lägger mycket tid och resurser på att förhindra illegal kopiering och spridning av deras produkter. Ett exempel på detta är SOPA (Stop Online Piracy Act), ett lagförslag som gör det möjligt för amerikanska myndigheter att söka ut och ta till rättsliga processer mot webbplatser och internetjänster som bryter mot federal upphovsrättslagstiftning. (112th congress, 2011-2012)

I och med att internet fortfarande är en relativt fri miljö kommer all mjukvara som kan laddas ner och användas även att finnas för nerladdning. Att vara dynamisk och anpassningsbar när man presenteras med förändring ser jag som en förutsättning för att nå framgång. Problemet i att motarbeta piratkopiering genom att förhindra kopiering är att du genom detta försöker tvinga världen runt omkring dig till att anpassa sig efter dina önskemål. Det är tyvärr ett vanligt misstag i vårt samhälle och då inte bara inom media och underhållning utan överhuvudtaget. Om vi vill fortsätta att röra oss framåt måste vi inse att det bästa sättet att förhålla sig till förändringar är att anpassa sig själv, inte att försöka ändra på alla andra. Genom att ge ut mitt spel gratis är extern (och annars illegal) kopiering och spridning av produkten inte något som skadar mig varken som utvecklare eller utgivare.

3.1.4. FPS-spel som tävling

Jag upplever genren First Person Shooter som en väldigt tacksam utgångspunkt för ett tävlingsinriktat spel. Grundkonceptet är extremt enkelt, både för spelare och för åskådare; man skjuter och blir skjuten. Det hela är väldigt svart och vitt, vilket gör att man enkelt kan implementera ett poängsystem som låter både spelare och åskådare följa spelets gång. Runt dessa två idéer byggs en tävlingsmiljö, där spelare sätts i någon form av arena där de slåss mot varandra. Spelaren med högst poäng i slutet av rundan vinner. Det är ett system som gör det lätt, även för den oinsatte, att titta på och förstå vem som vinner och vem som förlorar.

Det finns många olika system för hur man tar poäng, ett populärt alternativ är Capture the flag. (CTF). I CTF delas spelarna upp i lag och varje lag har en bas med en flagga. Lagen tar poäng genom att bära hem fiendelagets flagga, men för att ta poäng måste ditt lags flagga vara säker i sin bas. Detta leder till att lagen måste klara av att använda både en offensiv och defensiv spelstil.

I den här typen av spel finns det ofta en hel del funktioner som styr spelets gång eller vilka förutsättningar en spelare har. Många spel låter spelaren starta med ett simpelt vapen och placerar ut extra resurser på banan i form av till exempel vapen, ammunition och bandage. Andra spel låter spelaren bestämma sin utrustning innan spelet börjar eller vid speciella tillfällen under sessionen.

På senare tid har ytterligare funktioner vuxit fram. En populär modell är att låta spelaren låsa upp utrustning och uppgraderingar genom någon form av virtuell valuta, som oftast tjänas ihop genom att spela och ta poäng. Detta är ett väldigt bra system för ett F2P spel, då man som gratisspelare kan investera tid i att låsa upp nya funktioner eller alternativt betala en mindre summa för att få tillgång till samma funktioner omedelbart. Spelare som väljer att betala sparar tid och de som väljer att spela gratis får investera lite mer av sin tid i spelet, men har fortfarande tillgång till samma innehåll.

Oavsett hur poängsystemen ser ut eller vilka funktioner spelet och banorna introducerar så har spelet en väldigt enkel kärna. Du och alla andra har sina vapen och de används för att uppnå ett visst mål. I slutändan räknas poängen och den individ eller det lag som gjort bäst ifrån sig vinner.

Det är viktigt att komma ihåg att alla delar tillsammans utgör den slutliga upplevelsen och att alla delar av ett system måste fungera i harmoni för att resultatet ska bli lyckat. Varje del presenterar dock en rad unika problem som alla kräver unika lösningar. Det är därför viktigt att från början vara klar med hur spelet ska marknadsföras och användas och att spelet i sig tar hänsyn till dessa externa variabler.

3.2. Frågeställning

Vilka problem ställs man inför som nybliven spelutvecklare under utvecklingen av ett gratis, tävlingsinriktat spel?

3.3. Syfte

Syftet med denna undersökning är att identifiera och undersöka design- och implementationsproblem som uppstår kring tävlingsinriktade FPS spel samt att titta lite närmare på hur ett spel bör anpassas för att fungera på en gratismarknad.

3.4. Tidigare forskning

Inom mjukvaruutveckling är det vanligt att produktioner överskrider sin budget och sina deadlines och läggs ofta ned. Om man jämför detta med andra typer av produktioner, till exempel konstruktion av byggnader med hjälp av etablerad ingenjörsvetenskap, så syns en stor skillnad. Det tycks kanske märkligt, men bristerna hos mjukvaruutveckling kan enkelt förklaras genom insikten att utveckling av mjukvara är ett hantverk och inte en ingenjörskonst.

Det finns inga riktiga vetenskapliga metoder för att i förväg avgöra hur ett program ska konstrueras. En programmerare lär sig sitt yrke genom övning och erfarenhet. En ingenjör kan däremot använda sig av vetenskapliga regler och matematiska formler för att i förväg konstruera virtuella modeller.

Så trots att det finns ramverk och modeller för hur vi ska bedriva våra projekt så är 'Trial and error' till stor del inbyggt i vår mjukvaruutveckling idag. De ramverk som finns har uppstått genom att andra människor har provat sig fram och nått en slutsats, precis som medeltidens arkitekter (till skillnad från dagens ingenjörer) fick prova sig fram för att sedan skapa ritningar och dela med sig av sina erfarenheter till sina lärjungar. (Buhner 2003, 1)

Det kan vara till stor hjälp att tidigt identifiera viktiga designelement i sitt spel och att sortera dessa i kategorier. Detta är en enkel fallundersökning av spelet 'Diablo'.

*Tabell 1, exempel på ett spels presentation i ett givet moment.
(Bethke 2003, 88)*

Display system
Terrain: Draw floors.
Terrain: Draw Isometric walls.
Terrain: Color cycling special effects for water and lava (tiles do not animate).
Terrain: Ghost walls when a character is located behind the wall.
Characters: Render and animate characters (2D sprites composed from 3D rendered models).
Game Objects: Colored outlines for interactive objects such as treasure chests, magical rings, monsters, and non-player characters in the town center.
Spell Effects: Display any one of a couple of dozen effects with dazzling animations and cool sound effects.
Menus: Display menu choices.
Movies: Display the intro and exit movies to the player.
Audio: Hear sound effects.
Audio: Hear music.
Audio: Hear voice-overs.

Här kan vi se viktiga designelement som har med presentation och feedback att göra. Designelementen delas upp i kategorier (Terräng, karaktärer etc) och de mer grundläggande funktionerna presenteras först, detta ger en lätt överskådlig bild av spelets komponenter.

Bethke (2003, 220) menar att innan man kan skapa en omfattande lista av komponenter och tillgångar så behöver man definiera och förstå spelupplevelsen i sin helhet, en så kallad '*walkthrough*' för hela spelet.

Mitchell (2012, 81) skriver i sin bok att en designer alltid måste ställa sig själv frågan: fungerar grafiken som utvecklas tillsammans med spelets tänkta visuella stil? (Med grafik refererar vi till karaktärer, rekvisita, omgivningar och gränssnitt.)

I de fall där ett spel utvecklas för en existerande *IP (Intellectual Property)* finns det redan en värld och karaktärer att fylla den med och spelets gameplay byggs sedan runt detta. Annars kommer konceptet för spelet först och sedan utvecklas miljöer och karaktärer för att matcha känslan man vill uppnå i spelet. I vilket fall bör man alltid sträva efter samma slutresultat: en god balans mellan grafik och gameplay. (Mitchell 2012, 81)

Ett spel idag måste mer eller mindre ha ett tema och då inte bara ett grafisk sådant, utan även ett tema för gameplay. Det ger nya spelare en uppfattning om vad spelet handlar om och gör spelet mer enhetligt. Gameplaytemat bör vara direkt relaterat till spelets gameplay. Pirater och grekisk mytologi skulle kunna vara det grafiska temat, men det beskriver inte spelets gameplay. Om man istället formulerar spelets gameplay på detta vis: 'Det är ett fantasyspel i en värld full av monster och magi' skulle inte duga då det endast beskriver miljöer och features. (Oxland 2004, 55)

F2P-modellen har vuxit väldigt snabbt de senaste åren och det finns några olika variationer för hur utgivaren kan tjäna pengar på den. Vanligtvis kommer inkomsten från någon form av försäljning av virtuella tillgångar, såsom i Linden Lab's Second Life där virtuella föremål och fastigheter säljs för riktiga pengar. En annan variant är ett system där utgivaren agerar mäklare för försäljning av föremål mellan två parter. Ett exempel är Blizzard's Diablo 3 där spelare tillåts sälja sina föremål och guld för riktiga pengar till andra spelare, för denna tjänst tar Blizzard en liten del av vinsten. En annan tjänst som använder en liknande modell är Apple's iTunes. (Davis 2009, 227-228)

Jagex RuneScape och CipSoft's Tibia är båda helt gratis att spela, men för en månadsavgift får man tillgång till diverse bonusar, till exempel områden som endast betalande kunder får tillgång till. (Davis 2009, 228)

4. Tillvägagångssätt

4.1. Arbetsmodell

Produktionen utgick ifrån en beställning, beställningen i sin tur utgick ifrån en förundersökning samt en produktbeskrivning. Förundersökningen kan ta flera former. I det här projektet bestod den av en undersökning av problemområdet samt en analys av beställningen. Detta resulterade i en central frågeställning med en motiverande bakgrund, ett syfte som förklarar undersökningens relevans och efterforskning vars uppgift var att skapa en utgångspunkt för projektet genom att undersöka liknande arbeten. Produktbeskrivningen konstruerades därefter dels för att fylla det syfte som beskrevs i förundersökningen och dels för att undersöka frågeställningen.

Det första jag gjorde i projektet var att bryta ned produkten i mindre delar eller komponenter. Varje komponent innehåller en självständig funktion. En komponent skulle till exempel kunna vara den del av servern som lagrar informationen från en spelsession och varje komponent ska kunna kopplas ihop med en annan komponent i systemet. Genom att hålla sig till detta ramverk fördelas arbetet mellan utvecklare och genom att specificera den indata som komponenten ska hantera samt den utdata som den ska producera spelar det ingen roll hur komponenten fungerar. Komponenterna fungerar som pusselbitar och så länge som de passar med de andra bitarna kommer pusslet att gå ihop.

Varje funktion bryts ned till komponenter, vars utdata steg för steg går igenom tills det att hela komponenten finns beskriven (som en ritning). Låt oss kalla dessa steg för milstolpar. En teknisk lösning kan sedan tas fram av utvecklaren och genom att ha tydligt definierade milstolpar kan projektledaren lätt se hur projektet förhåller sig till tidsplanen.

4.1.1. Produktionsmål och produktbeskrivning

Jag förstod tidigt att det är problematiskt att definiera ett slutmål för den här typen av produkt, eftersom onlinespel ofta följer en utvecklingsmodell där de fortsätter att uppdateras kontinuerligt över en väldigt lång tidsperiod efter släpp. Ett sätt att tackla detta problem är att jobba utifrån riktlinjer eller kanske en frågeställning snarare än ett specifikt mål.

Om jag som spelutvecklare vill skapa en tävlingsmiljö som är balanserad och intressant för både spelare och åskådare kan jag lätt testa nya idéer genom att ställa frågan: "Skulle implementation av denna idé föra spelet närmare det jag försöker skapa?". Nu finns det helt plötsligt underlag för en diskussion. Varje element av idén kan diskuteras och undersökas utifrån denna frågeställning. Fördelar och nackdelar kan identifieras och genom att iterera på denna process kan en idé som kanske delvis passar in utvecklas till något nytt som helt passar in. I de fall där idén inte fungerar får man konkreta argument för varför det inte fungerade och man kan anpassa sin arbetsprocess i framtiden utefter det.

Jag började alltså med att skapa en produktbeskrivning för spelet, till att börja med såg den ut som följer:

- ▲ UT-inspirerat arena FPS
- ▲ Tävlingsinriktat
- ▲ Väldigt stiliserad och 'enkel' grafik, till stor del svart och vitt, lite gråskala
- ★ Multiplayer
- ~~★ Möjligtvis RPG element, såsom förmåga att bygga en egen loadout och kanske låsa upp/uppgradera delar av sin loadout genom att man får XP i sina spel.~~
- ▲ ~~Mycket av den grafiska upplevelsen kommer genom effekter såsom projicering, ljussättning/skuggning, partikeleffekter mm~~

Jag märkte dock tidigt att några av dessa punkter inte skulle få plats i den här produktionen och heller inte var helt relevanta för den frågeställning jag satt upp, de två sista punkterna försvann.

Därpå bröt jag ner produktbeskrivningen i komponenter. Punkt ett, spelet i sig, skulle vara ett simpelt FPS med deathmatch där två eller fler spelare tävlade mot varandra. Högre tävlingselement brydde jag mig inte om i det här utvecklingsstadiet, jag behövde lägga grunden för spelet innan det blev dags att tänka på det.

Jag valde ett enkelt grafiktema då mina kunskaper inom grafisk modellering och animering är extremt begränsade, förhoppningen var att hinna lite längre genom att hoppa över texturering.

När det gällde multiplayer blev jag tvungen att skapa en ny produktbeskrivning, eftersom den nya servern blev ett såpass stort projekt. Jag hade tre krav:

- ⤴ Stöd för minst 8 spelare
- ⤴ Central server som håller reda på sessionen och bestämmer vem som behöver få vilken information för att spara så mycket bandbredd som möjligt.
- ⤴ Ingen direkt kommunikation mellan användare, all information ska hanteras av servern innan den skickas vidare.

4.2. Produktion

4.2.1. Produktionens mål

Målet med undersökningen var alltid att hitta problem och att undersöka designval genom att jobba mot en given produktbeskrivning. Målet var aldrig en färdig produkt. Jag utforskade, för mig, nya områden i nästan alla skeden av min produktion vilket ledde till att det blev svårt att utforma en tidsplan. Eller kanske snarare, det blev svårt att hålla sig till de tidsplaner som utformades under projektets gång. Resultatet blev att jag istället hittade områden som jag ville undersöka närmare och nu till slut har jag en uppsättning komponenter som rent mekaniskt fungerar var för sig, men behöver lite mer tid för att sättas ihop till en produkt.

4.2.2. De stora problemen

Redan under förundersökningen var jag tvungen att beskära min frågeställning en hel del och nu i efterhand skulle jag nästan vilja påstå att jag borde ha minskat ner den ytterligare. Alternativt ha valt ett bättre verktyg för att undersöka problemet. Jag blev tvungen att lägga väldigt mycket tid på att bygga min spelserver, något som jag från början trodde skulle gå ganska fort men som visade sig mer tidskrävande än beräknat. Hade jag till exempel använt Windows Live eller byggt spelet i UDK som redan innehåller stöd för onlinespel skulle jag kanske hunnit längre i min produktion, men å andra sidan känner jag att den centrala servern är viktig för den här typen av spel. Vad Windows Live och UDK har är i princip ett peer-to-peer (P2P) nätverk, där en klient agerar server. Problemet som kommer med detta är att man blir väldigt beroende av spelvärdens uppkopplingshastighet. Med en central server kan man mer eller mindre garantera en stabil uppkoppling vilket leder till en bättre spelupplevelse i längden.

4.2.3. Första problemet, paketstruktur

Under vidareutvecklingen av min server stötte jag på tre stora problem. Övergången från 2D till 3D innebar att alla metoder för paketering av vektorer behövde skrivas om, vilket snarare var tidskrävande än svårt.

4.2.4. Andra problemet, analoga kontroller

Systemet hade tidigare använt digitala kontroller (8 riktningar) men behövde nu stödja analoga kontroller (teoretiskt oändligt antal riktningar), vilket medför vissa problem. Först och främst ökar trafiken mellan klienterna och servern avsevärt, eftersom spelaren kommer att kunna byta riktning oftare än vad som tidigare var möjligt. Då det inte är praktiskt att skicka positioner varje gång de ändras skickas istället riktningar och klienterna får använda den informationen för att själva lista ut var de andra spelarna är någonstans. Ett sådant system för med sig en viss felmarginal eftersom kommunikation över ett nätverk är relativt långsam och opålitlig. Jag beslutade att det behövdes ett nytt system som kontrollerar att positionerna i den lokala klienten stämmer med positionerna på servern.

4.2.5. Tredje problemet, serverstruktur

Det som i slutändan orsakade mest problem var övergången från ett P2P system med stöd för två spelare till en dedikerad central server med stöd för åtta eller fler spelare. Detta var ett problem som jag underskattade enormt mycket och som visade sig kräva helt nya sätt att paketera och synkronisera information. Tidigare visste varje klient om när en ändring skedde hos den andra spelaren, då information skickades direkt mellan spelarna när ändringar skedde. Denna lösning blir exponentiellt mer och mer opraktiskt ju fler spelare som är anslutna till sessionen, speciellt när en eller flera spelare inte alltid behöver veta vad alla andra spelare gör. Till exempel när två spelare står på varsin sida om en vägg, i ett sådant fall räcker det att servern är medveten om var de båda spelarna befinner sig och endast upplyser dem när de båda är i en position där de kan se varandra. Bandbredd är den mest begränsade resursen som finns när man utvecklar ett onlinespel, varje byte som skickas räknas när resultatet av en match kan avgöras av minsta lilla lag.

För att lösa detta problem designade jag ett system där servern själv kan bestämma när och till vem data ska skickas. Varje klient kan dessutom begära all sessionsdata när som helst. Ett problem som däremot inte är löst ännu är hur servern ska hålla reda på geometrisk information, då servern inte har tillgång till scenen.

Någon form av abstrakt representation av banan som spelas vore användbart. En lösning jag tänkte på var att kartlägga ytor som gränsar till varandra och anta att spelare som befinner sig på samma yta kan se varandra. Det för dock med sig nya tekniska problem och inför dessutom svårigheter för bandesignern.

4.2.6. Fjärde problemet, spelmotorn Sunburn 2.0

En annan aspekt av projektet som jag underskattade var omställningen till Sunburn 2.0 motorn. Den bygger förvisso på XNA och det är lätt att förstå hur den fungerar när man ser färdiga verk. Något som däremot inte var riktigt lika lätt var att designa lösningar till nya problem som exemplen inte tagit upp. Eftersom min insikt i motorn och dess funktioner är begränsad kunde jag inte utforma mina lösningar på samma sätt som jag tidigare hade gjort i XNA.

En hel del tid gick åt till att designa och utveckla ett system för att kontrollera entiteter i scenen. Den lokala spelaren använder en kontrollfunktion som är inkluderad i den inbyggda fysikmotorn, jag var dock inte säker på hur jag skulle översätta dessa kontrollinmatningar till min representation i de andra spelklienterna.

Mitt första försök till fjärrstyrning var samma lösning som jag tidigare använt i mina nätverksspel. När spelaren rör på sig skickas riktningen till alla andra spelare och positionen skickas när man stannar, för att kompensera för eventuellt lag eller borttappade paket. Problemet med detta var de nya, analoga, kontrollerna. När spelaren vrider sig bara några grader byter man riktning hundratals gånger, vilket leder dels till mycket tyngre nätverkstrafik men också till större skillnader mellan det som syns på fjärrdatorn och det som syns i den lokala klienten.

Detta försök ledde dessutom till en ny insikt; eftersom den lokala spelaren använder fysik och krafter för att röra på sig behöver även de fjärrstyrda spelarna någon form av kontrollklass för att återskapa samma beteende utan att behöva göra samma komplexa uträkningar. Jag försökte anpassa klassen som används för lokal styrning till att styra fjärrspelare, men det visade sig vara en krånglig och onödig och till slut även opraktisk process. Fjärrspelarnas positioner har redan blivit uträknade i deras respektive lokala klienter och att räkna ut allt igen hos fjärrdatorn kändes inte bra.

Man kan kringgå mycket av denna problematik genom att justera krafterna som påverkar karaktärerna på ett sådant sätt att interpolationen mellan minimal hastighet och maximal hastighet sker omedelbart, snarare än över en kurva. Man kan då använda riktningen spelaren färdas i och helt enkelt flytta entiteten med dess maxhastighet, då den aldrig kommer att befinna sig i någon gråzon.

För tillfället är detta lösningen jag har valt att använda: Spelaren skickar positions- och riktningdata till servern som i sin tur lagrar och skickar informationen till de andra spelarna. De i sin tur använder informationen för att manipulera scenobjektens positioner, kontroller görs emellanåt för att kompensera för skillnader mellan klienterna. Det är inte ett perfekt system och jag har fortfarande inte hittat en lösning som hanterar de analoga kontrollerna på ett helt tillfredsställande sätt.

Trots problemen jag haft med att lära mig motorn har den bidragit med väldigt många viktiga funktioner till mitt projekt. Den största fördelen har varit sceneditorn som låter mig bygga scener och spara dem i ett eget format. Programmet kan sedan ladda olika scener med hjälp av filen. Detta betyder att jag aldrig behöver bygga de statiska delarna av scenen i kod, utan kan fokusera på de dynamiska delarna och spelmekaniken.

Utöver sceneditorn innehåller Sunburn 2.0 även en kraftfull renderingsmotor med stöd för deferred rendering, ljussättning och skuggning, koncept som jag har jobbat med förut och är bekant med men som skulle tagit väldigt många timmar att göra själv. En stor del av min undersökning handlade om speldesign. Om jag hade gjort spelet i XNA hade jag kanske kommit lite längre vad gäller gameplay, men jag hade behövt offra många av dessa funktioner, för att inte nämna sceneditorn. Jag känner att mitt beslut att använda Sunburn 2.0 gynnar både mig och spelet i längden och att jag nu står närmare mitt tänkta resultat (mätt i arbetstimmar) än jag skulle ha gjort om jag byggt spelet i XNA.

4.2.7. *Alternativet, spelmotorn UDK (Unreal Development Kit)*

Ett alternativ som jag övervägde var att bygga spelet i UDK, en motor som mer eller mindre är byggd för den här typen av spel. Den största fördelen där hade varit att nätverksstöd är inbyggt i motorn, man får dessutom en fin lobby och snygga menyer. På många sätt hade det nog hjälpt mig med min undersökning, men ett par problem fick mig att välja Sunburn 2.0 istället.

Ett problem med UDK är att programmeringen sker i deras egna skriptspråk, UnrealScript. Detta innebär två problem; jag har väldigt lite erfarenhet i UnrealScript, vilket innebär en viss risk. Det är oftast lättare att lära sig en motor som använder ett språk man är bekant med, än att lära sig både en motor och ett språk samtidigt. Logiken kommer vara densamma, men om man inte kan orden blir programmeringen en väldigt långsam och plågsam process (speciellt när man jobbar mot en kort deadline).

Det andra problemet är att UnrealScript är en väldigt specialiserad kunskap, till skillnad från C# eller C++ som går att använda utanför Sunburn 2.0 och XNA i mer konventionell programmering. Ett mål jag alltid försöker tillfredsställa i mina projekt är att de saker jag lär mig ska gå att använda på flera sätt. Jag anser att specialisering är ett tvåeggat svärd, speciellt när man befinner sig i början av sin karriär.

Man blir kanske mer åtråvärd för vissa arbetsgivare, men samtidigt begränsar man sina val. Håller jag mig till C# får större kontroll över mitt programmerande och kan välja att designa mina projekt på ett sätt som maximerar mitt lärande samt min flexibilitet som programmerare. Det är kanske inte av jättestor relevans för projektet i fråga, men är av allra största relevans för mig som person. Så länge jag bara svarar till mig själv och inte tar betalt av en beställare kommer mitt personliga lärande alltid att stå i centrum.

4.3. Design

4.3.1. Gameplaytema

Grundidén var alltid att skapa ett spel som i botten var ett FPS med fokus på tävling och lagspel och att lägga funktioner som gav spelet sin egna unika säljpunkt och charm ovanpå denna grund. En svårighet med att utveckla ett spel för tävling är att spelet måste vara väldigt färdigt både mekaniskt och estetiskt innan det kan fylla sin roll.

En e-sport måste ta hänsyn till såväl åskådarna som spelarna och spelet måste vara lätt att följa samtidigt som det måste fungera felfritt under en spelsession.

Om servern inte håller ihop, klienten är dåligt optimerad eller kontrollerna inte fungerar etc. skulle det fungera dåligt som tävling. Som utvecklare av både tävlingsmiljö och redskap anser jag att man har en skyldighet att se till att allting fungerar, innan man officiellt annonserar att spelet är färdigt.

4.3.2. Grafiskt tema

Vad gäller grafiken och det grafiska temat var min ursprungliga idé en väldigt minimalistisk svartvit stil, med dynamiska element som introducerade vissa färger. Poängen var att skapa en miljö som var väldigt lätt att se och ta till sig, med starka kontraster och tydliga konturer.

Vartefter jag arbetade efter denna modell insåg jag mer och mer att min grundidé inte riktigt räckte, stora delar av resonemanget bakom idén fungerade men det behövdes lite mer. Jag ville att spelet skulle se nästan tvådimensionellt ut och komma till liv när man rörde sig i miljön, ett resultat jag hoppades uppnå genom att bara belysa vissa element i scenen. Effekten fungerade dock inte riktigt som jag hade hoppats.

Om jag istället använde en teknik som kallas cel-shading (eller toon shading) skulle jag kunna få ett väldigt tvådimensionellt utseende som dessutom är konsekvent genom hela scenen. Istället för att belysa olika objekt på olika sätt kan jag på så vis applicera samma ljussättningsteknik överallt. Detta är dock fortfarande i planeringsstadiet och är något jag kommer att få undersöka vidare i framtiden.

4.3.3. **Karaktärer**

Kravet jag ställde för karaktärerna var att de måste skilja sig kraftigt från den övriga dekoren. När en annan spelare befinner sig i spelarens synfält ska man direkt kunna se dels att det är en spelare, men också vad för utrustning denne bär på. Genom att ha tydliga former och framförallt konturer på olika typer av utrustning kan man informera andra spelare vad för roll och funktion man själv fyller. En spelares lag kan på samma sätt representeras med färger, eftersom det är viktigt att kunna differentiera mellan vän och fiende.

4.3.4. **Former**

Former, konturer och färger spelar alltså en viktig roll. I spel såsom World of Warcraft tas detta på stort allvar, vid design av ny utrustning tar man alltid hänsyn till klassen som föremålen är avsett för. Man vill till exempel att en spelare lätt ska kunna skilja en präst från en krigare, eftersom denna information avgör hur spelaren förhåller sig till sin motståndare. Blizzard använder den här designen i alla sina spel, även Diablo 3 och Starcraft 2 och det tycks vara något många tar hänsyn till.

Den här kunskapen har haft ett stort inflytande på min karaktärsdesign. Eftersom jag siktar på ett grafiskt tema som till stor del består av monokrom karaktärs- och dekordesign blir det konturerna på karaktärer och utrustning som informerar andra spelare vad de ser. En idé jag har lekt med är att låta spelare använda både skjutvapen och svärdliknande vapen, vilket skulle kräva väldigt tydliga skillnader dels mellan vapnen i sig men även i karaktärernas hållning och rörelser.

4.3.5. **Färger**

Att helt utelämna färgvariation i karaktärs- och bandesignen känns mindre och mindre gångbart ju längre arbetet fortskrider. Det finns mycket att vinna genom färgläggning av vissa element hos karaktärerna och dekoren, utan att det grafiska temat och känslan som önskas blir lidande. Karaktärerna kommer behöva lagfärger, vapen och speciella delar av den övriga utrustningen kan behöva sticka ut lite extra och interaktiva föremål på banan behöver också synas lite tydligare.

Planen var från början att introducera lite färg i miljön genom bl.a. partikeleffekter och ljus och jag anser fortfarande att detta är en bra idé. Jag har dock inte fått möjlighet att undersöka detta ännu, det befinner sig fortfarande i idéstadiet. Dessutom har jag börjat undersöka sätt att introducera lite färg i karaktärsdesignen, lagfärger om inte annat.

5. Resultat och diskussion

5.1. Resultat

Målet med undersökningen var att identifiera problem som kan uppstå av till exempel oerfarenhet, inkompetens, bristande planering, tekniska begränsningar etc. och att dokumentera dessa problem, för att senare kunna dra lärdomar genom analys och diskussion. Under min produktion stötte jag på ett flertal problem.

5.1.1. Produktion

- ⤴ Valet att jobba med en ny motor ledde till problem, tack vare brist på erfarenhet och kunskap gick mycket tid åt till inläring av grundläggande kunskaper och misstag vid utveckling av nya system.
- ⤴ De delar av programmet som hanterar paketering och tolkning av paket krävde en konvertering från 2D till 3D vektorer.
- ⤴ Mitt system hade tidigare endast implementerats i spel med digitala kontroller, introduktionen av analoga kontroller visade sig vara mer problematiskt än väntat.
- ⤴ Omstruktureringen från ett P2P system till en central spelserver krävde stora ändringar i källkoden och ledde till oförutsedda problem. Det gamla systemet för delning av lokal data fungerade inte alls i den nya designen och någon bra lösning har ännu inte hittats.

5.1.2. Design

- ⤴ Utformning av ett tema för såväl gameplay som grafik har visat sig vara ett rörligt mål, jag upptäcker ständigt problem med mina idéer och tvingas uppdatera min designfilosofi.
- ⤴ Karaktärsdesign kändes lättare, i alla fall på ett abstrakt plan. Produktionstiden jag hade till mitt förfogande visade sig inte vara tillräcklig för implementation.

5.2. Diskussion

5.2.1. Produktion

Som tidigare nämnt överskrider de flesta IT-projekt sin tidsgräns, vilket var något jag fick erfara i denna produktion. Jag lyckades spara mycket tid inom vissa områden (fysik, kollision, ljussättning/skuggning) genom att använda Sunburn 2.0 istället för XNA. Detta medförde dock problem i och med att jag saknar tidigare erfarenhet, ett misstag här var avsaknaden av en genomgående kunskapsinventering. Jag hade självklart en viss uppfattning om vad jag kunde, men att se igenom kunskaperna jag har till mitt förfogande och ta hänsyn till dessa när jag estimerar tidsåtgången för varje steg i produktionen är något jag kommer lägga mer tid på i framtiden.

Konvertering och uppdatering av gamla system är något som bör tas på stort allvar och inte underskattas, har jag märkt. När jag såg över kraven för nätverkskommunikationen i början av projektet såg det ut som en (relativt) snabb åtgärd, men de ändringar jag planerade från början förde med sig många svårlösta problem. I efterhand tänker jag att det kunde ha varit en bra idé att undersöka existerande ramverk för nätverkskommunikation. Jag tror det kan vara värt att alltid undersöka fler alternativa lösningar innan produktionen startar, även i de fall där man känner sig säker på det första alternativet.

Planen var alltid att ge ut mitt spel gratis och om möjligt föra diskussioner med människor som provar spelet för att bestämma hur spelet bör växa. Jag tänkte skapa en hemsida och ett forum där man kan ladda ned och diskutera spelet samt hitta information och nyheter. Produktionen har tyvärr inte kommit till den punkten än, men planen kvarstår och kommer med all sannolikhet att realiseras inom en snar framtid.

5.2.2. Design

Ett konsekvent designtema är som sagt viktigt för helhetsupplevelsen, jag märker dock mer och mer att min bristande erfarenhet och de insikter jag kommer fram till under arbetets gång successivt avslöjar brister i min design. Detta gör att jag ofta blir tvungen att gå tillbaka och revidera mina beslut, en tidskrävande process som skulle behöva tas hänsyn till i tidsplaneringen.

Karaktärsdesign kändes lättare och min originalidé fungerar såvitt jag kan se till stor del även nu. Vissa ändringar har gjorts i detaljerna, men överlag är det former och konturer som är viktiga att hålla sig till. Så länge jag håller mig innanför de regler jag satt upp för proportioner, rörelser och former kan jag göra ganska mycket och skapa viss variation i detaljarbetet.

5.3. *Slutdiskussion och konklusion*

Som spelprogrammerare måste man programmera mycket för att bli bättre i sitt yrke, man måste kontinuerligt tillämpa sina kunskaper, hitta problem i sin arbetsmodell och göra sitt bästa för att analysera och lära sig av dessa problem. Målet med mina projekt är inläring samt personlig och professionell utveckling, produktionen i sig ser jag mest som ett verktyg eller ett hjälpmedel för mina självstudier.

En programmerare lär sig som jag nämnt tidigare sitt yrke genom att prova sig fram och att studera under andra. Man måste programmera mycket och man måste lära sig att analysera sina resultat, lyckade såväl som misslyckade, dra lärdom och applicera dessa lärdomar i framtida projekt. Det är därför essentiellt att driva projekt där man kan utmana sig själv och utvidga sina gränser.

Min idé med det här projektet var att påbörja den här typen av projektdrift i en miljö där jag fortfarande har tillgång till god handledning, en stabil studiemiljö och bra feedback. Jag såg det som en möjlighet att istället för att genomföra samma typ av produktion som jag gjort tidigare, för att försöka koppla ihop mina tidigare studier här på BTH med det liv som väntar mig efteråt.

Kravet för lärande är konstant och kommer inte att upphöra i och med att jag blir klar med den här utbildningen, utan medför snarare ett paradigmskifte. Jag kommer att vara i konstant behov av utbildning och vidare lärande. Den stora skillnaden är att jag nu kommer vara ansvarig för mitt eget lärande till hundra procent.

I det här projektet försökte jag till så stor del som möjligt styra mitt eget arbete, mina mål och mitt lärande som träning inför mina framtida självstudier och för att ge mig en utgångspunkt när jag kliver ut ur skolan.

6. Ordlista

Arena FPS eller shooter

En typ av FPS-spel som uppfyller vissa krav:

- Alla spelare börjar med samma förutsättningar.
- En spelares arsenal bestäms inte av någon förutbestämd modell, utan all utrustning finns i arenan och är del av de resurser som spelarna måste slåss om.
- Dessa resurser placeras på ett sådant sätt att de styr spelarnas rörelser och aktivitet i spelsessionen.
- Det finns inga fördefinierade roller eller specialiseringar för spelarna; alla börjar matchen med identiska förutsättningar och resurserna man samlar under spelets gång skapar olika möjligheter.

Cel-shading

Cel-shading är en ljussättningsteknik med vars hjälp man kan få en 3D-modell att likna mer klassisk 2D animering.

First person shooter (FPS)

Ofta förkortat till *FPS*, är en datorspelsgenre där bildskärmen motsvarar spelfigurens synfält och där skjutvapen spelar en stor roll.

Free to play (F2P)

Ett spel som är gratis att införskaffa och spela. Bygger ofta på att spelare köper saker i spelet för verkliga pengar, till exempel föremål, bonusar eller virtuell valuta.

Intellectual Property (IP)

Ett juridiskt koncept som refererar till intellektuella skapelser till vilka det finns särskilda upphovsrättsregler.

Lag/lagg

Latens (från engelska *latency*), även kallat lag. Är ett tekniskt uttryck som beskriver tidsfördröjningar eller svarstider i ett nätverk.

Peer-to-peer

Ett P2P-nätverk (peer-to-peer network) är ett icke-hierarkiskt datornätverk av sammankopplade noder som inte kommunicerar enligt klient-server-modellen

Walkthrough

Förklarar alla upplevelser som spelaren utsätts för i spelet. Skulle till exempel kunna täcka hela kampanjen i ett spel som har en sådan. För en online shooter går man igenom en hel match, från skapandet av matchen till det att poängen visas i slutet.

7. Källförteckning

112th congress (2011 – 2012). Stop Online Piracy Act (H.R. 3261), The Library of Congress.
<http://thomas.loc.gov/cgi-bin/bdquery/z?d112:h.r.3261:>

Bethke, Erik (2003). Game Development and Production, Wordware Publishing, Inc.

Brenda Brathwaite och Ian Schreiber (2008). Challenges for Game Designers, Cengage Learning.

Buhrer, Hans Konrad (2003). Software development: what it is, what it should be, and how to get there, ACM SIGSOFT Software Engineering Notes Volume 28 Issue 2, March 2003 Page 5.

Davis, Steven (2009). Protecting Games, Charles River Media.

Giuditta De Prato and Jean Paul Simon (2011). Games and Innovation Research Seminar 2011 Working Papers. Chapter 3: Updating Business Models: Innovation through Online Games, Page 16.

Mitchell, Briar Lee (2012). Game Design Essentials, SYBEX.

Oxland, Kevin (2004). Gameplay and Design, Addison Wesley.